The iPhone Apps. A Digital Culture of Interactivity

Flückiger, Barbara

Originally published at:
Flückiger, Barbara (2012). The iPhone Apps. A Digital Culture of Interactivity. In: Snickars, Pelle; Vonderau, Patrick. Moving Data: The iPhone and the Future of Media. New York: Columbia Universtiy Press, 171-183.

CHAPTER 11

The iPhone Apps

A Digital Culture of Interactivity

BARBARA FLUECKIGER

PATRICK COLLISON, who in his own words is a "hacker, pilot, student at MIT, cofounder of Auctomatic," and "lover of waffles," certainly can be seen as prototypical of certain first-generation developers of iPhone apps—the whiz kids.[1] Self-taught, he started to program software at an early age. When Patrick was seventeen, he founded his own company, Auctomatic, with his younger brother John and sold it two years later for an exorbitant sum to the Canadian company Live Current Media. During the winter of 2007 he programmed the iPhone app Encyclopedia, an offline version of Wikipedia that allows almost all of Wikipedia's online functions, including the use of links between different entries and in 2010 was offered in eighty-three languages, including Chinese, Hindi, and Vietnamese.

In a broad sense, Patrick Collison is an example of a "digital native." But he is more than that, given that "digital natives" need not, by definition, be creative in developing tools; they need only be highly literate in exploiting predefined structures. As I will argue, following the writings of the media philosopher Vilém Flusser, it would be a gross misunderstanding to believe that a technology brings forth mental structures or abilities. More often than not, as my investigations into the history of technological change have revealed, thought models develop in a wider cultural context before they result in new technologies, which in turn influence patterns of behavior and thus the "wiring" of thoughts. Accordingly, this chapter discusses aspects of the evolution and properties of a digital culture that led to the development of the iPhone as a multidimensional tool with functionalities far exceeding the making of phone calls. App development is a striking example of a technological achievement with a massive impact on the social and cultural structures that govern its use. After starting with some observations regarding the practice and history of iPhone app development, I will investigate the epistemological aspects of digital encoding in the main part of this chapter. At the end, I will connect these investigations and deduce certain general insights pertaining to the cultural and mental consequences of this new technology. This chapter addresses the role of iPhone apps, as well as their development and distribution, within the framework of a technological history of media development in the digital domain. The present study, like other studies of technological innovation I have conducted in the past, may be described with Frank Beau's term of "technobole": an analysis that focuses on a technology to extract from it an understanding of its position in culture and society. In this view, a technology is not the source of teleological change, as technological determinism would assume, but rather a node in a far reaching network of scientific knowledge and cultural artifacts. More broadly, my approach can be seen as related to the body of work often subsumed under the label of actor network theory, which represents an antiessentialist, pragmatic view of sociomaterial processes and the history of knowledge.

The Sweet Solution

At the June 2007 Worldwide Developers Conference, Steve Jobs presented the iPhone to an excited crowd.[2] The idea of third-party apps was already present: "We have been trying to come up with a solution to expand the capabilities of the iPhone so developers can write great apps for it, but keep the iPhone secure. And we've come up with a very. Sweet. Solution. . . . An innovative new way to create applications for mobile devices . . . it's all based on the fact that we have the full Safari engine in the iPhone." What Jobs and Apple's Scott Forstall were talking about were Web-based apps running on a browser, which limited their possibilities for development. At that time, Jobs and Forstall had a mere eleven apps that they presented over and over, including the calendar, the address book, and photos, apps that continue to constitute the core block of preinstalled apps on the iPhone. In March 2008 there were approximately 1,000 Web apps available, and the situation improved further after that when Apple launched its Apple SDK (software development kit) platform, which gave third-party developers a sound basis for the development of apps. According to Steve Jobs's keynote address at the June 2008 WWDC in, ninety-five days after its launch the SDK had 250,000 downloads and 25,000 registered developers.

With the SDK—which Scott Forstall introduced on the day of its launch in a presentation entitled "iPhone Software Roadmap"—external developers were given an application programming interface and a variety of tools to make use of the internal architecture of the iPhone and its built-in devices. These included, for example, the localizer, which triangulates the position of the iPhone and connects it to a Google map, and the accelerometer, a three-axis device for the positioning of the iPhone in space that adjusts the screen to its vertical or horizontal position and allows the iPhone to be used as a controller for games.

The API consists of four main architectural layers: the core operating system, iOS, an adapted version of Apple's OS X; core services, such as CoreLocation for the development of location-aware apps (for example, to connect to nearby friends and find restaurants), the address book, and the SQLite database; media for the use of audio-visual content, core animation for the creation of layered animation, and the Open Graphics Library for Embedded Systems, a hardware accelerated interface for 3D graphics applications; and, finally, Cocoa Touch, the user-interface application framework that enables user control of content by touching the screen or by the use of the accelerometer or localizer. In addition, the SDK offered several tools, most importantly the Xcode integrated development environment to write code for a new project. This source code that controls a given application is usually hidden from users.[3]

Another tool is the Interface Builder, which facilitates graphic-interface design based on drag and drop. Developers can choose from a menu of predefined controls (buttons, switches, the wheel) or invent their own custom controls. "Cocoa Touch supports the model view controller paradigm of development," which also visually connects the view layer to the control layer. "Because it is a visual editor, you get to see exactly what your interface will look like at runtime," Forstall asserted in his "iPhone Software Roadmap" presentation.

The Interface Builder dictates standardized interfaces, providing a set of visual building blocks that ensures that every application developed by third parties fits into Apple's corporate design. All of these developer tools are supported by a range of extensions to test and debug applications, either by connecting the iPhone to a desktop computer or by running the apps on an iPhone simulator directly on a Mac. Yet first-generation developers such as Patrick Collison did not need such predefined structures. He wrote his Encyclopedia app six months before the App Store was even launched. As experienced hackers, these young people were able to gain their knowledge independently by investigating the iPhone and its operating system itself. As Collison put it in a private message, he "had to 'disassemble' the built-in apps to figure it out."

Binary Data Encoding and Digital Thinking

The term "digital" is often overused and overgeneralized; it does not differentiate the multifaceted phenomena that rely on digital code. Nevertheless, there is a basic property common to any form of digital representation, namely, the binary mode of data encoding. As a universal mode, this encoding process enables a variety of interactions with data that range from its transmission—the feeding of data in and to a variety of media—to its transformation—the processing of data by mathematical operations—and random access, which allows data to be accessed directly in a nonlinear fashion.

In 1988 Vilém Flusser published an essay called *Krise der Linearität* (*Crisis of Linearity*).[4] In my view, Flusser's essay remains the most valuable text for understanding the fundamental shift that digital data have brought about in our culture, society, and thought. Flusser's cryptic and idiosyncratically structured essay offers an analysis of the historical change in representing the world that has occurred through technologies ranging from the early cave paintings of Pech Merle to computer-generated images. Flusser addresses one of the most important aspects of representation, namely, the interaction between the underlying epistemological principles of a given representational technique—painting, alphanumerical texts, photographs, digital representations—and thought. While most scholars focus on the impact of technology on culture, Flusser turns this question upside-down and asks for the cultural foundation that arguably leads to a change in representational codes and in turn affects our thoughts, feelings, wishes, and imagination. While his observations mainly address digital representations, they also offer fundamental insights into the functions of digital data in general.

According to Flusser, painting emerged to orient a community toward future actions, such as organizing the chase of animals. Painting required the subject to stand back from the object not in terms of spatial distance but in terms of mental abstraction. To communicate a singular perception in an intersubjective way, the painter had to resort to his inner state—memory and imagination—to convey the outer world in a universal conventionalized language. In a second step the symbols that resulted from this technique were organized in a linear manner to move from the still ambiguous connotative meaning of pictures to the more denotative form of texts to satisfy the societal need for a more rational and thus more reliable communication. This change in encoding from a two-dimensional plane to a one-dimensional line brought a shift to linear thinking through a teleological model of development and rational cause-and-effect chains, and this shift remained at the center of Western culture for a long time. Yet the alphabetical code of texts still lacked a precise instrument to investigate imaginations and thus required an extension in the form of a numeric code. As Flusser establishes, the emerging alphanumerical code was in itself deeply contradictory: "While letters unravel the surface of an image into lines, numbers grind this surface into points and intervals. While literal thinking spools scenes as processes, numerical thought computes scenes into grains."[5]

If we follow this analysis, it is clear that binary code, which is fundamentally informed by its numerical and mathematical foundation, challenges traditional notions of linearity in the most radical manner not by representing the world as two-dimensional pictures or one-dimensional texts but by breaking down phenomena into clouds of zero-dimensional points. In contrast to the traditional forms of representation that produced effigies, which in turn served as models for future actions, synthesized digital images (i.e., computer-generated images) produce models that in turn might result in objects. That is, imagination predates perception, an observation for which Flusser introduced the German term "*Vorbilder*" to mean both models and "pre-images," or antetypes. One may describe Flusser's model of development as a cybernetic feedback loop in which cultural and technical forces enhance or correct each other. But this model still relies on a linear understanding of history, which in Flusser's view is based on the discovery of deficiencies in society that call for solutions. So he in fact combines an underlying teleological model with a circular or even dialectical structure.

Flusser's thoughts are clearly based on a materialist view that we may summarize best with Marshall McLuhan's catchphrase that "the medium is the message." Flusser's analysis shows its potential in relation to iPhone apps when we reflect on the impact on iPhone users' perceptions that the zero-dimensional pointlike mode of binary code and its deep roots in the mathematical domain of numbers will have. It is here that we can connect

Flusser's philosophical insights to the development and the distribution of iPhone programs.

Transformation and Mutation

In contrast to the hardware of established media technologies in the electromechanical domain—film, TV, radio, sound recordings—digital media comprise two layers: the hardware that houses the functions and the software that describes and thus generates the functions. While earlier analogue techniques involve an intricate connection between the flow of information and its material foundation, in the digital domain the two elements are completely separate, with the binary code defined arbitrarily by a protocol for encoding and decoding digital information. These complementary actions are at the foundation of every digitization and thus of every software program.[6] These insights go back to Nelson Goodman's distinction between autographic and allographic processes, a distinction central to William J. Mitchell's investigation of visual truth in the post-photographic era.[7] While autographic processes such as painting comprise only one stage from production to finished object, allographic processes such as musical notation systems require two stages, first, the writing of the notes on paper and, second, the notes' interpretation, which transforms the written text into a process accessible to the auditory system, for which it was intended from the beginning.

In a similar fashion, digital code or software is no more than a notation system for a future display in the planned domain. With the allographic system, however, comes another specific property of digital media objects. "Traditionally, musical scores, literary texts, and other specifications of allographic works have had final, definitive, printed versions," Mitchell notes. "The act of publication is an act of closure."[8] This does not apply to digital code as allographic because such code remains open to mutation. This openness blurs the distinction between producers and consumers since consumers may have access to the data, either directly on the level of the coding system or indirectly with the help of interpreting software that offers a graphical user interface. Transformation—and thus programmability—as well as interactivity are core properties of digital culture.

Both transformation and interactivity need an interface, and this is where Apple has a huge advantage in interface design based on the GUI and a developer-friendly API. In 1984, when using a computer still required active knowledge of computer code, Steve Jobs introduced the GUI into the

Macintosh universe. This GUI offered users a metaphorical surface consisting of graphical symbols that linked the world of computers to traditional office environments. It has become the industry standard since Microsoft's Windows operating system gained ascendancy in the 1990s. In fact, the GUI is the most important step for the spreading and democratization of home computers and—in their wake—of mobile devices such as laptops and now the iPhone, because it connects the opaque site of the binary encoding and control of the hardware with a transparent, intuitively accessible surface that is aesthetically pleasing.

Attempts have been made to simplify human-computer interaction since the beginning of computer history. A light gun was developed at MIT in the 1950s to allow the direct addressing of individual points on the cathode ray tube monitor of MIT's Whirlwind computer. Later in the same decade, the light pen was introduced as an input device to communicate with the computer. The single most important invention toward establishing a GUI was Ivan Sutherland's "Sketchpad." Presented in 1963, it was the result of his Ph.D. thesis at MIT and offered users possibilities to create, transform, and store objects on the computer. There was also a zoom feature to enlarge the view.[9] A few years later, Doug Engelbart from Stanford Research Laboratory developed the mouse and presented it to the scientific community. The mouse was easier to use than the light pen, and it spread in connection with the windows and icons metaphors that were to provide the building blocks of the GUI. At the Palo Alto Research Center of Xerox, Alan Kay then developed the windows style of the GUI even further—and from there it found its way into the Macintosh operating system.

When Steve Jobs presented the iPhone in 2007, usability and interface design were central to its potential success. The interface was key to differentiating the product from competing smartphones that—according to Jobs—were smarter than ordinary cell phones but not easy to use. There is, to draw again on the abstract theoretical discussion of transformation and mutation, an essential shift from systems dominated by hardware with mechanical buttons and controls, as in smartphones like the BlackBerry, to a system controlled in large measure by a flexible surface entirely open to any software design. In the two-stage allographic mode this means that the interpreter of the notation system has a much broader range of interaction open to his or her needs, provided by the vast possibilities of designing controls as pure graphical elements. With the touch screen operated by Cocoa Touch, this tiny computer taps the essence of transformation and mutability owed to basic binary encoding. The touch screen is also where the approximately 250,000 iPhone apps developed to date find their place to unfold myriad specific tools across an almost unlimited spectrum.

Steve Jobs thus seems to have been right when he stated in 2007 that with this design Apple was years ahead of its competition. Since 2007, most competitors have followed Apple's route—similar to how Microsoft adopted the GUI in the 1980s. Hardware-wise, the iPhone offers a variety of physical subsystems to which the apps can be connected, thereby further widening their range. In addition to the core processor there are many additional functionalities, such as the media processors for audio-visual content, the animation core for animated content, the Open Graphics Library, the accelerometer for the control of the iPhone itself in 3D space, and the localizer, which makes use of the GPS system, as well as the telephone and Internet-access capabilities and of course the touch screen. These hardware modules enable apps to connect these functionalities in individual ways, based on the transformation capabilities of the binary code and on random access.

An application can thus be understood as a translation device, enabling communication between the user interface and the hardware by a specific protocol. And this is where the second layer of interface comes into play, put into practice by Apple's SDK as the application programming interface. Much as Jobs had stressed the iPhone's touch screen, Scott Forstall said of the SDK that as a development environment it was years ahead of the competition in the mobile device market on its launch in 2008. In a Twitter post the same day, Patrick Collison agreed that this statement was not "marketing SPEAK"; instead, he stated that the SDK was the main advantage that would "cause them [Apple] to win the smartphone war" while at the same time bemoaning "the end of an era of reverse engineering. All those late nights spent pouring over . . . assembly."

Apple's primacy generally stems from the company's long history in the creation of various development interfaces. "They simply did a better job of creating the tools for allowing developers to create *good* software—that looks good, and works well. This is one of those things that I think people outside of the software world usually miss—the extent to which the nuance and tiny detail of implementations have a big impact in a way that's very hard to quantify," Patrick Collison told me in an e-mail in August 2010.

Random Access

A second, arguably even more important consequence of the pointlike structure of digital data is random access. Random access relies on digital code's distinct values to allow direct addressing of the individual numbers, thereby facilitating nonlinear connections between individual points in the data space to create network structures. A variety of practical applications arise from this. The first is the Internet, with billions of URLs that can be retrieved by billions of users. A second is the hypertext structure of Web-based documents, offering texts with layers to be navigated freely and hypermedia with text, graphics, and audiovisual media such as QuickTime files to be connected in myriads of individual ways. And a third application is the connection between surface elements such as graphic icons, the controls such as the touch screen, and the accelerometer with the hardware elements by means of software.

In this context one might recall McLuhan who—with startling prescience in 1964—predicted the emergence of the global village as a network structure in society that would implode space. To be sure, McLuhan attributed this change to electricity, not to the universality of digital code or to its point-like form of representation. McLuhan also confused the electric and the electronic. While light is electric, every device that implies a control of capacity or resistance or alters voltage or current, is electronic. Only electronic devices enable electric ones to become carriers of information, such as the radio. Electronic devices, in turn, have to be separated from digital ones, insofar as they still rely on an analogue relation between signal and encoding and thus do not employ binary coding. These distinctions are crucial as they separate different stages in the development of technology. Mechanical, electric, and electronic devices still belong to the domain of linearity as they produce processes that unravel in time. It is only with the digital that the mathematical form of representation shifts to spatially distributed spots that allow for random access. However, there were nonlinear systems even in the electro-mechanical age, such as card indices used in libraries to organize data. Even books can be used in a nonlinear manner based on an index that invites a nonlinear reading.

This historical irregularity accounts for the aforementioned observation that predecessors of a technological change can almost always be found. Furthermore, it documents a cultural need to overcome hard-edged linear or even nonlinear strategies and devices, which prompted Vannevar Bush to write his famous article, "As We May Think," in 1945, in which he states: "The human mind . . . operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain. . . . Man cannot hope fully to duplicate this mental process artificially, but he certainly ought to be able to learn from it."[10] Moreover, in his original definition of the term hypertext in the 1960s, Theodor H. Nelson proposed an interactive screen as an appropriate device for associative navigation.[11] Associative patterns, then, are at the very core of random access. We could even state that random access not only mirrors mental processes, as Bush noted, but also—by the very act of selection—breaks the world down into bits and pieces. Every fragment that emerges out of this process is a node in a new network built by the user. Associative mechanisms in thinking are thus perpetually enhanced and lead to change that challenges the traditional Western model of linear progress. Vilém Flusser embedded this notion in a broader cultural context when he identified modern conceptions of the world as proposed by quantum theory to be a precursor to this fragmented, nonlinear style of thinking. As a consequence, he saw the dissolution of the subject "in a collective psychic field, from which we emerge like temporary bubbles, acquire some information, process, share, to submerge again." Moreover, "we are immersed in an undulating field of culturemes, from which the individual cultures emerge through computation, just to blur again."[12]

Conclusion: Network Structures and App Development

The interactive culture of iPhone apps is in many ways intimately intertwined with the kind of network structures that Flusser calls the "collective psychic field." This holds true, of course, for every social activity on the Internet, including every Web 2.0 activity—from participation in forums to open knowledge sources such as Wikipedia to social networks. However, it is also a new phenomenon for a technological development to arise from this structure. Interestingly enough, it was exactly this vision that guided one of the masterminds of the Web, J. C. R. Licklider, a psychologist who provided several ideas for the use of computers. In his 1968 paper "The Computer as a Communication Device" (written with Robert W. Taylor), Licklider proposed that "collaboration in creative endeavor [could gain] critical mass" by connecting people over computer-aided communication. "Take any problem worthy of the name, and you find only a few people who can contribute effectively to its solution. Those people must be brought into close intellectual partnership so that their ideas can come into contact with one another."[13]

This is exactly what happens in the programming of iPhone apps, where developers form a community to share ideas and get advice on solving individual problems. In this way, a company can outsource both its development work and the risk of failure that is intrinsic to every advance in technology. Companies can make use of masses of specialists all over the globe. Many developers come from countries such as Russia, India, or China and would likely have difficulties gaining access to institutionalized structures. The introduction of the SDK marked a shift from a free hacker culture in the first months of the iPhone's existence to a strictly formalized and institutionalized one. With the introduction of the App Store in June 2008, Apple gained maximal control over the distribution of apps and thus strengthened its influence further. While the first few months with the hacker system reflected possibilities of transformation and mutation in an innocent state, exploring in depth the native properties of the digital culture, the restrictions that followed had a severe effect on this free-floating state.

First of all, these restrictions implied a standardization of the apps as expressed in the Interface Builder. Second, and more importantly, these restrictions brought closure to an initially fully open system. This closure occurred on different levels, not least on the level of the text, that is, the software and its code for each individual app. Once an app has passed the strict evaluation process to be distributed in the App Store, it is closed and then is open to mutation only by the original developer and by hackers who use illegal ways to gain access. Marxist scholars such as Jean-Louis Comolli would argue that this is the classic story of the capitalist system appropriating innovations that come from the margins of society. Astonishingly, this view is also expressed in Chris Anderson's article, "The Web Is Dead: Long Live the Internet," published in *Wired*—a magazine hardly known for its critique of hegemonic ideology—which investigates the change of the Internet from an open web to a controlled distribution channel for proprietary apps. "This was all inevitable. It is the cycle of capitalism. The story of industrial revolutions, after all, is a story of battles over control. A technology is invented, a thousand flowers bloom, and then someone finds a way to own it, locking out others."[14] As Anderson argues, it is we who give these companies their power, because we prefer to get our software solutions from controlled sources and not from browsing unreliable download pages.

Does this form of institutional ownership imply that the idea of a digital culture of interactivity is dead? Yes, in part. While it is true that this step implies standardization and closure, the system is open to a broad movement of masses who could collaborate in this venture. Only with the launch of the App Store did it become possible for developers to become entrepreneurs. According to a survey from Flurry in March 2010 one in five developers are start-up enterprises.[15] These small companies were launched in order to develop either apps for mobile devices or independent software that could be transformed into apps.

Yet there are still independent individuals able to implement ideas based on their everyday experience, such as meeting up with friends, writing grocery lists, and controlling moods. In their spare time, individuals have even developed games and music-playing interfaces such as MooCowMusic. Some of the most successful apps, such as the music identification program Shazam or Loopt, a localization program based on GPS, also started their development long before the iPhone was introduced. One may wish to tell a story that follows the David versus Goliath narrative, with an individual hacker like Patrick Collison fighting the corporations. From an economic point of view, Apple is certainly a capitalist venture operating on a global scale. But beyond the exploitation of a mass of individual developers sharing their ideas and products, thus amplifying the iPhone's commercial success, the collaboration partly outweighs the capitalist pattern in operation. It is certainly indebted to the openness and universality of digital's binary encoding, with transformation and random access as its core properties.

Notes

1. See http://collison.ie (15 February 2011).

2. Videos of these keynote speeches are available on iTunes at http://itunes.apple.com/us/podcast/apple-keynotes/id275834665 (15 February 2011).

3. See http://developer.apple.com/iphone/library/referencelibrary/GettingStarted/URL_Tools_for_iPhone_OS_Development/index.html (15 February 2011).

4. Vilém Flusser, *Krise der Linearität* (Bern: Benteli, 1988). The English translation *Crisis of Linearity* is available at www.scribd.com/doc/26525368/Volume-1-Issue-1 (15 February 2011).

5. Ibid., 21.

6. See Nelson Goodman, *Languages of Art* (Indianapolis: Hackett, 1968); Malcom Le Grice, *Experimental Cinema in the Digital Age* (London: BFI, 2001); and Mark J. P. Wolf, *Abstracting Reality. Art, Communication, and Cognition in the Digital Age* (Lanham, Md.: University Press of America, 2000).

7. Goodman *Languages of Art*; Nelson Goodman, *Ways of Worldmaking* (Indianapolis: Hackett, 1976); and William J. Mitchell, *The Reconfigured Eye: Visual Truth in the Post-Photographic Era* (Cambridge, Mass.: MIT Press, 1992).

8. Mitchell *The Reconfigured Eye*, 51.

9. Ivan Sutherland, "A Man Machine Graphical Communication System," *Proceedings of the AFIPS Spring Joint Computer Conference* (Washington, D.C. 1963), 329–46.

10. Vannevar Bush, "As We May Think," *The Atlantic* (July 1945), http://www.theatlantic.com/magazine/archive/1969/12/as-we-may-think/3881/ (15 February 2011).

11. George P. Landow, *Hypertext 3.0: Critical Theory and New Media in an Era of Globalization* (Baltimore, Md.: Johns Hopkins University Press, 1992), 3.

12. Flusser, *Krise der Linearität*, 32–33.

13. J. C. R. Licklider and Robert W. Taylor, "The Computer as a Communication Device," *Science and Technology* 76 (April 1968): 29.

14. Chris Anderson, "The Web Is Dead. Long Live the Internet" *Wired*, 17 August 2010.

15. For a discussion, see www.appleinsider.com/articles/10/03/15/iphone_app_store_still_offers_level_playing_field_for_developers.html (15 February 2011).